ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 5, Issue 2, February 2018

An Empirical Analysis of Component Based Software Engineering and Critical Evaluation of Component Selection using Analytic Hierarchy Process: A Quantitative Approach

[1] Jogannagari Malla Reddy, [2] S.V.A.V. Prasad

[1] Department of Computer Science & Enginering, Lingaya's University, Faridabad, Haryana State, India

[2] Department of Electronics & Communication Enginering, Lingaya's University, Faridabad, Hayana State, India

Abstract: **- The Component Based Software Engineering (CBSE) is a sub discipline in Software Engineering, which would build the complex systems with integration of reusable prebuilt components. The reusable components are available in the Commercial of the Shelf (COTS), which are developed by third parties. The reusable concept reduces the development time, budget and human efforts in the software process. The requirement engineer performs crucial role in examining and selecting the proper component based on multiple criteria from the COTS. Analytic Hierarchy Process (AHP) provides the effective quantitative approach, which tackles such critical Multi Criteria Decision making problems. The intent of the paper is to evaluate and select the proper component from COTS using AHP, which leads to the customer satisfaction, reduces ambiguity, and prevents the software failure.**

Index Terms- **Analytic Hierarchy Process, Component Selection, Decision Making, Integration, Pair-wise comparison.**

I. INTRODUCTION

The software technology is an emergent technology which made the world as digital. The Software Engineering is a systematic, disciplined and quantifiable approach to develop the software products within the budget, time and with more quality. Lack of quality leads to unsatisfaction of the customer, insignificant cost, time and human efforts to the developer and vendor image damaged. The faulty systems can't fulfil the functional, non-functional and domain requirements of the customer. The Software Industry is the youngest compared to other industries and now it is nucleus for other industries. The other industries have profitable from reusable components such as electronic components on circuit board, vehicle parts replaced by components manufactured by different manufacturers. The concept is that standard interfaces allow for interchangeable, reusable components. In the words of the Roger.S.Pressman, 1977,"Industry is moving towards component-based construction, most software continues to be custom built". Today the software is also moving towards the components based construction, where reusable components can be simple such as push buttons, list boxes, combo boxes, radio buttons, scrollbars and check boxes etc. This technology is used in many application domains such as desktop graphical and web based applications [12]. The components based software development emphasizes the construction of software system with reusable components which are already developed and available at commercially of the shelf (COTS) [5]. The philosophy of the disciplines states that "buy and don't build", which reduces the time and human efforts [13]. The components are more specifically service providers is in the abstracted form than the classes of Object Oriented Systems[2]. The commercial off the shelf is pool where readymade software is available for sale to the general public which is developed by the third parties. The COTS consists of wide verity of component applications such as Office documentation, Statistical and Chemical Analysis, and Accounting Packages. These components facilitate a range of features and options to fit into the needs of versatile customer demands which support the built in configuration. Further, the component quality is assured in construction of software system. The aim of component domain engineering is to find the proper functional, behavioural and data components from the stored Capability Knowledge Base or reuse libraries (COTS). The domain engineering performs various activities such as Analysis, construction, and dissemination of components for existing and future use. More efforts are needed for proper component selection, verification and its integration from COTS. The selection of suitable component from the COTS repository is a complex and critical task; it plays a crucial role in the success of final system. Selection of the component depends on multiple evaluation criteria; criticality may arise in selecting the suitable component, because criteria's may be in conflict with each other. The Analytic Hierarchy Process is a Multi-Criteria Decision Making method [11] proposed to tackle the complex component selection problem. The selection of suitable component from COTS can avoid the ambiguity, misunderstanding and unsatisfaction of stakeholders and prevents the failure of software. The rest of the paper is organized as follows. The Section 2 illustrates the related literature of Component Based Software Engineering and methodology of Analytic Hierarchy Process. The Section 3 describes the overview spectrum of Component based software development and its various phases. Section 4 states the research methodology of Analytic Hierarchy process for component selection process. Finally a discussion about the future directions and inferences is presented in section 5.

II. RELATED WORK

Over the years a good number of researchers focused on the domain of component based software development and they highlighted various key issues. Their taxonomy can be useful to extend the knowledge in the components selection using Quantitative Approach. The appropriate approach used at right time helps expected quality outcomes in the software development.

* Bhupender Yadav et al [1], presented the research paper on various adequate testing techniques and its difficulties in Component Based Software Development. They highlighted the role of testing in quality based component software artifacts.

* Karambir Singh et al [2], identified the object oriented system properties in component based software products with wide literature survey.

* Evangelous Triantaphyllou and Stuart H. Mann [3], presented a paper on Quantitative Approach on AHP. They examined the experimental and computational issues involved in the engineering applications with the usage of AHP

* Suryani Ismail et al[4], characterized the Software component reusability for CBSE. They conducted survey on reuse practitioners of software at University of Malaysia. The findings of the metrics used to quantify the reusable component.

* Sharanjit Singh et al[5], emphasized the interface binding, risks and deficiencies of component based software engineering. They proposed the RAM process model for elimination of risks in CBSE.

* Ishita Verma[6], emphasized the design, construction and applicability components with its advantages and disadvantages in Component Based Software Engineering

* Ivica Crnkovic, et al[7], analyzed the various phases of Component Based Software Development Life Cycle with case study of company.

. * Javed Ali Khan, et al [8], published a research paper on various requirement prioritization techniques and proposed the flexible requirement prioritization method among existing, which is fault tolerant.

* Javed Ali Khan, et al [9], stated that, the Analytic Network Prioritization is the best method among the seven software requirement methods with case study.

* Thomas. L. Satty [11], derived the principles and Philosophy of Analytic Hierarchy Process for Multi Criteria Decision Making problems.

III. THE VARIOUS PHASES OF COMPONENT BASED SOFTWARE DEVELOPMENT

The domain of Component Based Software Engineering [CBSE] is an extended knowledge of Software Engineering reuse concept [2]. Software Engineering principles, methods and tools will be used in similar way in the CBSE. This approach extends the inheritance concept used in Frame-Based Systems, and imports the nuggets from several areas of Artificial Intelligence, in particular Compositional Modeling, Terminological Reasoning and Ontological Engineering. The domain engineering finds the proper functional, behavioural and data components from the stored Capability Knowledge Base (CKB) and reuse libraries. The components are developed for open market, for reuse that will be used in many different configurations and situation, many of them not foreseen at the time of development [10]. The component developers promote the components as black boxes without source code [7].

Domain Engineering focuses three major activities like Analysis, Construction and Dissemination of components for existing and future. Component Based Software development consists the roles of identification, verification and forming the relations between components to develop the final system. Maximum time will be taken for selecting, locating proper component than its integration in the software development.

The Component Based Software Development model starts with requirement specification, continues with selecting the suitable components by evaluation and concludes with integration of components into final system which is cost effective. There is a need of component selection and evaluation before it can be integrated into the system[7].

The various steps involved in the Component Based Software Development are as follows:

Requirement Analysis and specification: The requirement Analysis phase is a primitive phase of Component Based Software Engineering as traditional development. Requirement Engineer can analyze, negotiate the components which fulfil the requirements available in COTS and find, otherwise the risk factor have to be incorporated.

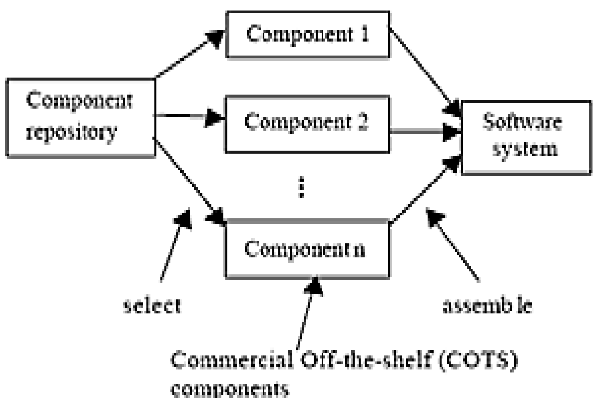

Figure.1. Component based Software Development

Software Design: The requirement specification phase and the design phase are strongly interrelated to the availability of components. Components are specific, complying with suitable component model and comfortable architectural framework. It is too complicated to use with another architecture to achieve the interoperability. The incompatibility has direct impact on architectural decisions. The Design process tightly coupled with its components [7]. Selection of suitable components which are comfortable with architectural design may be difficult to find, gaps may exist between the features of component and requirements. Component selection is a critical task depends on many criteria's and alternatives. There are some formal methods like 'Six Sigma' and informal methods like Experience-Based, Hands on-Trial Based and Customer Recommendation Based selection [13]. The design usually needs more iteration. At the end of the design phase, the components are selected based on criteria, to build the application and new components that are needed to be modified for interoperability.

Implementation and Unit Testing: As part of the implementation, the components are assembled, tested and debugging is made. The software components are in the form of glue-code which provides the interface with other components and implementation of new functions. The integration testing achieves the interoperability, which is more crucial than isolated testing of components [10].

System integration: Integration of components leads to deployment, which can form the final system of an application as a whole. The verification and validation is bit complex in CBSE compared to the traditional Software Systems due to invisible code. The defect malfunction lies on another component if not properly integrated. Component interface plays vital role to verify functionality of the component.

System Evaluation: The developed software system is not in satisfaction to the stakeholder in its life time due to process and technology changes that occur in its business environment [1]. Definitely it needs frequent maintenance to satisfy the client needs. In most of the cases the existing components are to be modified with minimum, otherwise upgrade with new component to the existing system.

IV. RESEARCH METHODOLOGY OF ANALYTIC HIERARCHY PROCESS FOR COMPONENT SELECTION AND PRIORITIZATION

The Analytic Hierarchy Process (AHP) is a Multi-Criteria Decision Making method which includes the concept of Weighted Scoring Method developed by the Saaty in 1980 and improved by Vargas in 2001. It has being applied for Decision Making in wide variety of management applications like banking, manufacturing, and education and the solutions are both objective and subjective. The AHP solves the various complex problems and derive the solutions with more powerful and in flexible manner [11].
Software component selection is a critical decision making task composed with multiple criteria's and alternatives. The software component selection is involved with Multi Criteria Decision Making process, the score of each alternative is derived using an appropriate AHP decision making algorithm. The method proves the effective solution for selection of best alternative with qualitative and quantitative [11]. The methodology is based on three principles: problem decomposition, comparative judgments and synthesis of priorities.

- The problem can be decomposed into hierarchy structure i.e. root level as goal of the problem, intermediate level as criteria's and node level as alternatives.
- The alternatives are compared with each other in certain criteria with pair-wise comparisons as per the Saaty's numerical scale of importance (1 to 9).
- Finally, calculate the aggregate performance value for each alternative and rate the alternatives according to the value on the numerical scale.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

A. Algorithm of Analytic Hierarchy Process

1 Define the problem and identify the criteria's and alternatives for each criteria.

2. Organize the problem into hierarchical structure

3. Construct the set of pair-wise comparison matrixes against each criteria. The element in an upper level is used to compare the elements in the level immediately below with respect to it. For each comparison matrix, evaluate the Eigen value, consistency index CI, consistency ratio CR, and normalized values for each criteria / alternative.

4. Use the priorities obtained from pair-wise matrix in global matrix. The scale for rating characteristics should be established and described in a precise way. This is done for every element. Then for each element in the level below add its weighted values and obtain its overall or global priority. The final value is used to make a decision about the objective.

B. Mathematical Representation

Step 1:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1,n} \\ a_{21} & a_{22} & \cdots & a_{2,n} \\ a_{31} & a_{32} & \cdots & a_{3,n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{n,n} \end{bmatrix}$$

$$a_{i,j} = 1, \text{for } i = j, a_{i,j} = \frac{1}{a_{j,i}} \text{for} a_{i,j} \neq 0$$

Step 2: Evaluate the nth root of product of each row.

Step 3: Find the Priority (p_k). The numbers are normalized (each row nth -root value) by dividing them with their sum.

Normalized Matrix- $$P = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ . \\ . \\ . \\ . \\ p_n \end{bmatrix}$$

$$A = \frac{A \times P}{P} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ . \\ . \\ \lambda_n \end{bmatrix}$$

$$\lambda_{\max} = \frac{\lambda_1 + \lambda_2 + \lambda_3 + \ldots. + \lambda_n}{n}$$

Consistency Index $$C_I = \frac{\lambda_{\max} - n}{(n-1)}$$

Consistency Ratio= $\frac{C_I}{R_I}$

Where R_I is Random Index refer table.2.

Step 4:

AHP formula for decision making:

$A^i_{AHP} = \sum_{j=1}^{N} a_{ij}\, w_j$, for i = 1, 2 … M ---- (1)

C. Analytical Study

The analytical research is conducted for component selection from the repository of Commercial off the Shelf (COTS). It is a complex activity and plays a vital role in the success of final software system. The component selection depends on the multiple evaluation criteria's; complexity may arise in selecting the suitable component, because criteria's might be conflict and nearer to each other. The Analytic Hierarchy Process proposed to tackle the critical component selection problem. The Component Selection can be evaluated based on its characteristics like Functionality, Usability, Interoperability, Cost and the pair-wise comparisons of AHP Methodology. The problem organized into hierarchical structure as the following figure.2.The pair-wise comparison inputs of AHP are collected from the eminent human perspectives of requirement engineers, designers, coders and the other stakeholders of the product through the questionnaire. The other pertinent inputs are taken from the software libraries

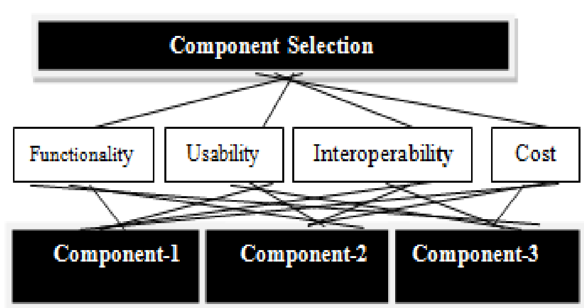

Figure.2. Hierarchical decomposition of Problem

The attention focused in the analytical study is to find out the suitable component based on its features and characteristics. Component feature weighted significance is more or less in between the multiple components. The pair-wise comparison matrix represents the corresponding judgment on scale of relative importance of the following table.1.

Table.1.Scale of Relative Importance (As per Saaty, 1990)

Weight	Definition	Explanation
1	Equal importance	Two activities in *equal* importance
3	Moderate importance	One activity *moderate* over another
5	Strong importance	One activity *strong* over another
7	Very strong importance	One activity *very strong* in practice over another
9	Extreme importance	One activity Extreme over another.
2,4,6,8	Intermediate values between two activities	Compromise is needed.
Reciprocals of above non Zero	If activity I has of above non nonzero numbers assigned to it when compared with activity j, then j has the reciprocal value when compared with it	

The next step in pair-wise comparisons, the corresponding maximum left Eigenvector approximated by using geometric means of each row [3]. Initially the Consistency Index (CI) can be estimated. This is done by sum of columns in the judgment matrix and multiply the resulting vector by the vector of priorities (i.e. approximated eigenvector) obtained earlier.The approximation of the maximum Eigen value denoted by λmax. Then, the C.I value measured by using the formula as CI = (λmax-n) / (n-1). Finally, the Consistency Ratio CR derived with CI value divided by Random Consistency index (RCI) as the table given below table.2.

Table. 2. Random Consistency Index on Matrix Size (As per Saaty, 2000)

Matrix Size(n)	1	2	3	4	5	6	7	8
RCI	0	0	0.58	0.90	1.12	1.24	1.32	1.41

Evaluate the CR, if the CR value is less than or equal to acceptable (0.10) which indicate a good level of consistency for decision making, otherwise inconsistency of judgments is seen within the respective matrix then the process to be reviewed, reconsidered and improved. The acceptable consistency helps to ensure decision making with more reliability.

The weights of importance of the criteria are also determined by using pair-wise comparisons. If the problem has M alternatives and N criteria, then the decision maker is required to construct N judgment matrices (each criteria) of order M*M and one judgment matrix of order N*N (for N criteria). Finally, the decision matrix final priorities denoted as Ai AHP .

$$A^{i}_{AHP} = \sum_{j=1}^{N} a_{ij} w_j, \text{ for } i = 1, 2, 3 \ldots M \text{ --- (1)}$$

D. Mathematical Calculations

Step 1:

The problem can be organized into hierarchical structure with goals, criteria's and alternatives.

Step 2:

Pair-wise comparison matrix of alternatives i.e. Component-1, Component-2 and Component-3 based on criteria of component features such as functionality Usability, Interoperability and cost.

(i) Criteria: Functionality:

$$A = \begin{array}{c} \\ \text{Comp-1} \\ \text{Comp-2} \\ \text{Comp-3} \end{array} \begin{bmatrix} \text{Comp-1} & \text{Comp-2} & \text{Comp-3} \\ 1 & 3 & 5 \\ 1/3 & 1 & 2 \\ 1/5 & 1/2 & 1 \end{bmatrix}$$

The Geometric Mean (GM) for nth row of the Matrix is as follows:

$GM_1 = (1 \times 3 \times 5)^{1/3} = 2.464$

$GM_2 = (1/3 \times 1 \times 2)^{1/3} = 0.874$

$GM_3 = (1/5 \times 1/2 \times 1)^{1/3} = 0.465$

Sum of the Geometric Mean (GM) = 3.802

Hence, the calculated Normalized priority vectors / weights (P) are as follows:

$P_1 = 2.464 / 3.802 = 0.648$

$P_2 = 0.874 / 3.802 = 0.230$

$P_3 = 0.465 / 3.802 = 0.122$

Now, the judgment consistency of pair-wise matrix is derived with the equation $\lambda = \frac{A \times P}{P}$

$$A \times P = \begin{bmatrix} 1 & 3 & 5 \\ 1/3 & 1 & 2 \\ 1/5 & 1/2 & 1 \end{bmatrix} \times \begin{bmatrix} 0.648 \\ 0.230 \\ 0.122 \end{bmatrix} = \begin{bmatrix} 1.948 \\ 0.690 \\ 0.367 \end{bmatrix}$$

IFERP connecting engineers... developing research

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 5, Issue 2, February 2018

$$\lambda = \frac{A \times P}{P} \begin{bmatrix} 1.948 \\ 0.690 \\ 0.367 \end{bmatrix} \div \begin{bmatrix} 0.648 \\ 0.230 \\ 0.122 \end{bmatrix} = \begin{bmatrix} 3.006 \\ 3.003 \\ 3.001 \end{bmatrix}$$

$$\lambda_{\max} = \frac{\lambda_1 + \lambda_2 + \lambda_3 + ... + \lambda_n}{n}$$

$$= \frac{3.006 + 3.003 + 3.001}{3} = 3.004$$

Then, the Consistency Index (CI)

$$CI = \frac{\lambda_{\max} - n}{(n-1)} = \frac{3.004 - 3}{2} = 0.002$$

(where n is the size of the matrix)

Finally, Consistency Ratio (CR)

$$CR = \frac{CI}{RCI} = \frac{0.002}{0.058} = 0.003$$

(where RCI value as per table 2)

So, the weights of pair-wise judgment matrix are consistent as per Saaty, (i.e. if CR < = 0.01)

The above pair-wise comparison and results of priorities (P), λmax., Consistency Index (CI), Consistency Ratio(CR) values mentioned the table.3 and diagrammatically represented in the pie graph figure.3.

Table. 3. Pair-Wise Comparison on Criteria of Functionality

Functional (Criteria)	Comp-1	Comp-2	Comp-3	Priority Vector
Comp-1	1	3	5	0.648
Comp-2	1/3	1	2	0.230
Comp-3	1/5	1/2	1	0.122
Total Priority				1.000
λmax. = 3.004,	CI = 0.002,		CR = 0.003	

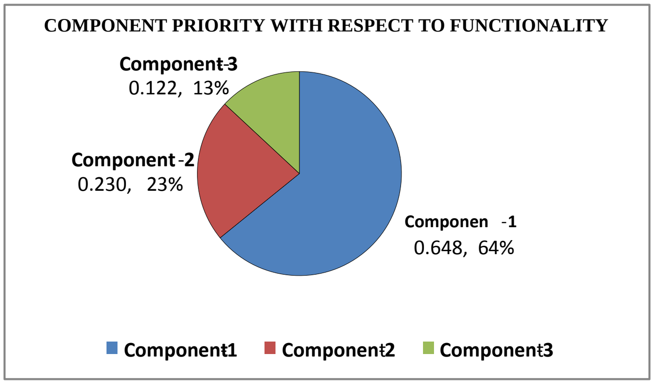

Figure.3. Priority vectors with respect to Functionality

Similarly,to find the other Criteria such as Usability, Interoperability, and Cost. The respective priorities (P), λ_{max}., Consistency Index(CI), Consistency Ratio(RI) values are mentioned in the table.4, table 5 and table 6.

(ii) Criteria: Usability

Table.4. Pair-Wise Comparison on Criteria of Usability

Usability (Criteria)	Comp-1	Comp-2	Comp-3	Priority Vector
Comp-1	1	3	5	0.637
Comp-2	1/3	1	3	0.258
Comp-3	1/5	1/3	1	0.105
Total Priority				1.000
λmax = 3.039,	CI = 0.019,		CR = 0.033	

Priority Vector of the Components with respect to Usability diagrammatically represented with pie graph in the figure.4.

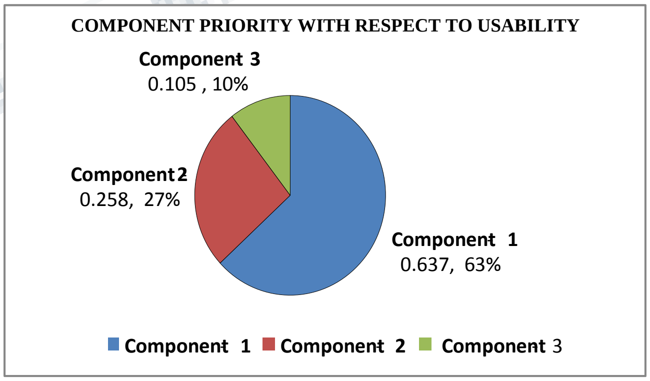

Figure.4. Priority vectors with respect to Usability

iii) Criteria: Interoperability

Table.5. Pair-Wise Comparison on Criteria of Interoperability

Interoperability (Criteria)	Comp-1	Comp-2	Comp-3	Priority Vector
Comp-1	1	3	9	0.671
Comp-2	1/3	1	5	0.266
Comp-3	1/9	1/5	1	0.063
Total Priority				1.000
λmax. = 3.029,	CI = 0.015,		CR = 0.025	

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

Priority Vector of the Components with respect to Interoperability represented with pie graph in the figure.5.

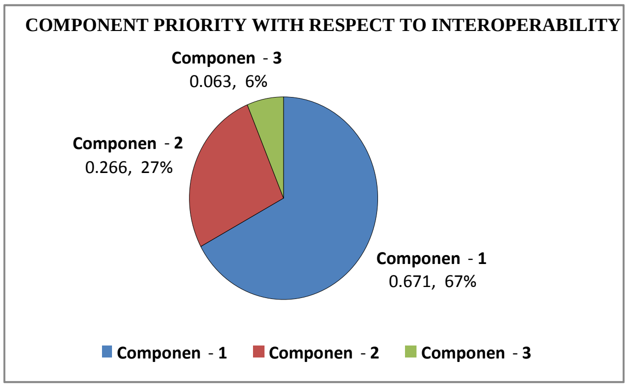

Figure.5. Priority vectors with respect to Interoperability

(iv) Criteria: Cost

Table.6. Pair-Wise Comparison on Criteria of Cost

Cost (Criteria)	Comp-1	Comp-2	Comp-3	Priority Vector
Comp-1	1	1/3	1/7	0.093
Comp-2	3	1	1/2	0.292
Comp-3	7	2	1	0.615
Total Priority				1.000
λmax. = 3.003,	CI = 0.001,	CR = 0.002		

Priority Vector of the Components with respect to Cost represented with pie graph in the figure.6.

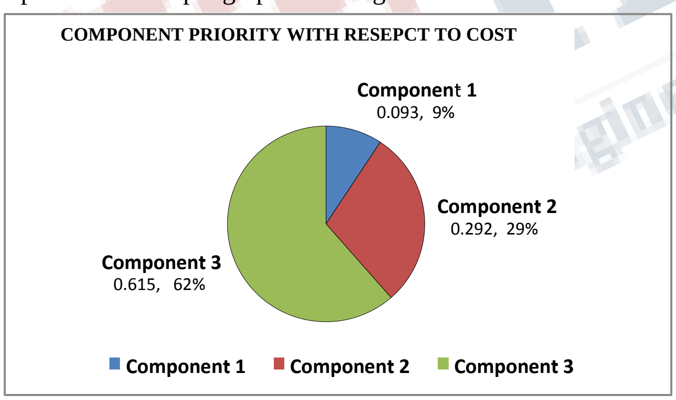

Figure. 6. Priority vectors with respect to Cost

Step 3:

The same mathematical process is continued to find out the priority vector (P), λmax, Consistency Index (CI), Consistency Ratio (CR) on the criteria importance of the component features i.e. functionality, Usability, Interoperability and cost. The results are shown in the table.7.

Table.7. Pair-Wise Comparison of Criteria Importance

Criteria	Functionality	Usability	Interoperability	Cost	Priority Vector
Functionality	1	3	3	1/3	0.240
Usability	1/3	1	1/5	1/9	0.054
Interoperability	1/3	5	1	1/3	0.158
Cost	3	9	3	1	0.548
Total Priority					1.000
λmax. = 4.260,	CI = 0.087,	CR = 0.096			

Priority Vector of the Criteria importance of component features shown in the figure. 7

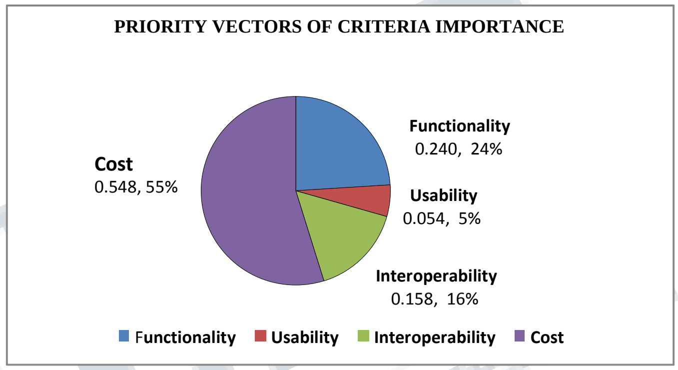

Figure.7. Priority vectors of Criteria Importance

Step 4:

The previous priority vectors (i.e. pair-wise comparisons against the criteria and criteria importance) (Step 2 & 3) are used to form the entries in final/global decision matrix. The problem has 3 alternatives and 4 criteria, then the decision maker is required to construct 4 judgment matrices (each criteria) of order 3*3 and one judgment matrix of order 4*4 (for 4 criteria) . Finally, the decision matrix the final priorities denoted as A^i_{AHP} and the results are mentioned in Table.8.

$$A^i_{AHP} = \sum_{j=1}^{4} a_{ij} w_j, \text{ for } i = 1, 2 \dots 3$$

Table.8. Component Selection Using Decision Matrix

Criteria	Functionality	Usability	Interoperability	Cost	Goal
	0.240	**0.054**	**0.158**	**0.548**	
Comp-1	0.156	0.034	0.106	0.051	0.347
Comp-2	0.055	0.014	0.042	0.160	0.271
Comp-3	0.029	0.006	0.010	0.337	**0.382**
Total Priority					1.000

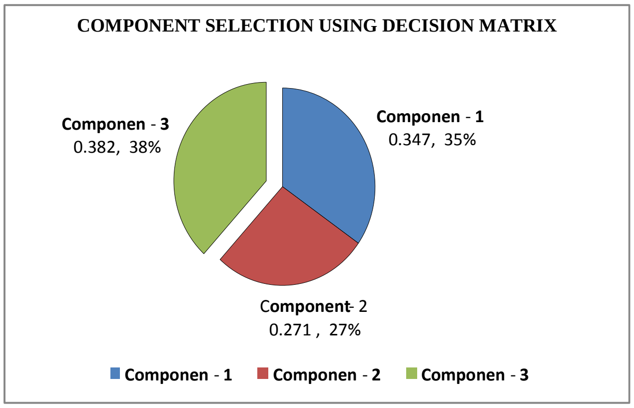

Figure.8. Component selection using decision matrix

The resultant (Componet-3) is an optimum component with required features that can satisfy the stakeholder requirements and prevents the software failure.

V. CONCLUSION

The Analytic Hierarchy Process is an effective Multi Criteria Decision Making Quantitative Approach for selecting and prioritizing the components from COTS according to the stakeholder criteria. The AHP can solve such problems in flexible manner and get the user satisfaction from all angles. Expert Choice is a semi-automated software that significantly contributes the features of AHP. The authors suggests that, some of the alternatives to be very close to each other, then the decision maker needs to be very cautious in giving inputs to the pair-wise comparisons. This problem mainly occurs with logical nature of software. The research in this area of Software Engineering is valuable and critical and may never end. Even tough, there is a need of extensive research in the area of component selection and prioritization with effective approaches.

VI. ACKNOWLEDGMENT

We sincerely, acknowledge to the faculty members, Department of Computer Science & Engineering, Lingaya's University, Faridabad for their support and cooperation during the research.

REFERENCES

[1] Bhupender Yadav and Anshu Yadav, "A Review on Component-Based Software Engineering and Testing", International Journal of Recent Advances in Engineering & Technology (IJRAET), Vol.4, Issue .2, pp. 127-129, 2016.

[2] K. Karambir Singh and P.K.Suri, "Technical Review: Inheritance of Component Based Software Engineering", International Journal of Advanced Research in Computer Science, Vol. 6, Issue. 5, pp. 95-103, May- June, 2015.

[3] Evangelos Triantaphyllou and H. Stuart Mann, "Using the Analytic Hierarchy Process for Decision Making in Engineering Applications: Some Challenges, "International Journal of Industrial Engineering Applications and Practice, Vol.2, Issue.1. pp. 35-44, 1995.

[4] Suryani Ismail, Wan M.N. Kadir, Noor Maizura Mohammad Noor and Fatihah Mohd, "Determining Characteristics of the Software Components Reusability for Component Based Software Development", Journal of Telecommunication Electronic and Computer Engineering, Vol. 9, Issue. 3-5, pp. 213-216, 2017.

[5] Sharanjit Singh, Amardeep Singh, Samson, Sandeep Singh, "Component Based Software Engineering", International Research Journal of Engineering and Technolog y(IRJET), Vol.5, Issue.5, pp. 448-454, May, 2016.

[6] Ishita Verma, "Study of Component Based Software Engineering," International Journal of Computer Science & Communication Networks, Vol.3, 2008.

[7] Ivics Crokovic, Stig Larsson and Michel Chaudron, "Component Based Development Process and Component Life Cycles," Journal of Computing and Information Technology, pp. 321-327, 2005.

[8] Javid Ali Khan, "Comparison of Requirement Prioritization Techniques to Find Best Prioritization Technique," I.J. Modern Education and Computer Science, Vol. 11, pp.53-59, Nov, 2015.

[9] Javed Ali Khan, "An Evaluation of Requirement Prioritization Techniques with ANP", International Journal of Advanced Computer Science and Applications, Vol. 7, pp. 419-428., 2016.

[10] Sushil Chandra Dimri, "An Innovative Model of Software Quality Assurance for Component Based Software Systems", International Journal of Emerging, Research in Managment and Technology, Vol.4, Issue. 7, pp. 216-219, July, 2015.

[11] TL. Saaty, "How to make a Decision: The Analytic Hierarchical Process," European Journal of Operational Research, Vol.48, pp. 9-26, 1990.

[12] Virendra Kumar and Shabina Ghafir, " Enhanced Software Development for Reuse Process Model in Component Based Software Engineering", International Journal of Computer Science and Technology, Vol. 3, Issue. 1, Jan- March, 2012.

[13] Loveleen Kaur and Hardeep Singh, "Software Component Selection techniques – A Review," International Journal of Computer Science and Information Technologies, Vol. 5, Issue. 1, pp. 3739-3742, 2014.